Générateur de graphes de dépendances

Stéphane Wouters

Janvier 2016

1 Description du projet

Ce projet a pour objectif de générer des graphes à partir d'un code source java. La génération est faite par une analyse statique du code source Java en utilisant l'arbre syntaxique abstrait (AST). La visualisation peut se faire dans un navigateur Internet.

Il est possible de générer deux types de graphes :

- Graphe d'appel au sein d'une classe
- Graphe d'appel d'une application

Lien GitHub (Sources) : https://github.com/Doelia/dependency-viewer

2 Description technique

Le projet comporte 3 composants :

- **Une application JAVA** qui permet de générer un graphe au format JSON en utilisant un AST. (Exporté en .jar pour une utilisation externe).
- **Un client web** qui permet de visualiser les graphes et de configurer des parametres.
  - Bibliothèque utilisée pour la visualisation : *Vis.io*
- **Un serveur en Golang** pour relier l'exécutable .jar au client web.

2.1 Documentation du code Java

Le projet Java est composé de 4 packages :

- *Package externals/* Permet la génération de l'AST via le pattern Visitor ;

- *Package builder/* Représente les entités Méthode et Type (classe) nécessaires à la gérération d'un graphe. Il propose toutes les méthodes nécessaires au stockage des ces entités et au calcul des dépendances entre elles ;
- *Package extractor/* Regroupe tous les outils nécessaires à la génération de graphes en fonction des objectifs voulus :
  - *Extractor.java* est la classe abstraite proposant les méthodes pour la construction du graphe ainsi que la conversion au format JSON de ces grahes ;
  - *ExtractorClasse.java et ExtractorClasse.java* sont la mise en pratique des cas voulus.
- *Package main/* contient la méthode main qui a le role de lancer les executions et de gérer les parametres d'entrée du programme

3 Utilisation

Le .jar est à disposition dans le répertoire et n'a pas besoin d'être recompilé.

Donner le droit d'exécution au .jar :

```
chmod u+x graph-generator.jar
```

Compiler le serveur golang :

```
ln -s golang/ast-linker \$GOPATH/src/ast-linker
cd golang/ast-linker
go build
```

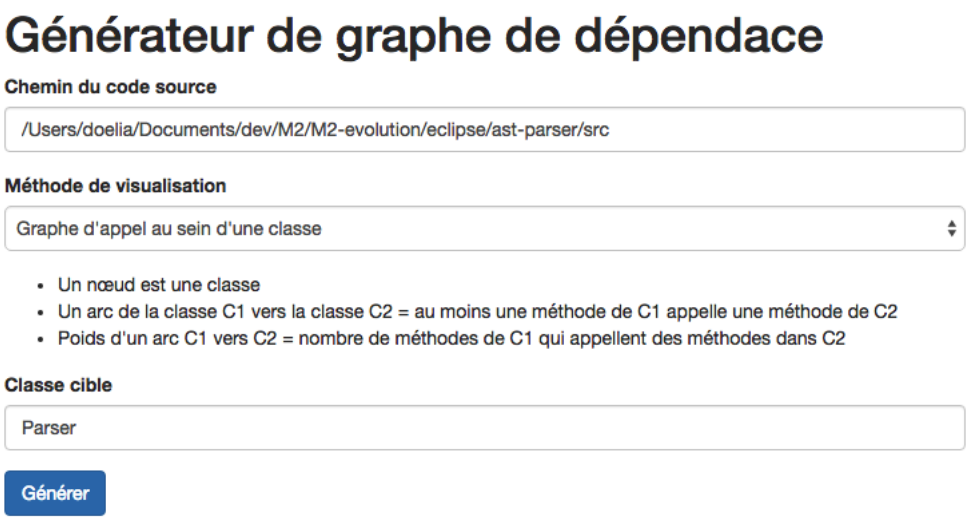
Lancer le serveur Golang en spécifiant le chemin absolue du JAR et le port HTTP voulu :

```
./ast-linker -port 2000 -jar \
/Users/doelia/Documents/dev/M2/dependency-viewer/graph-generator.jar
```

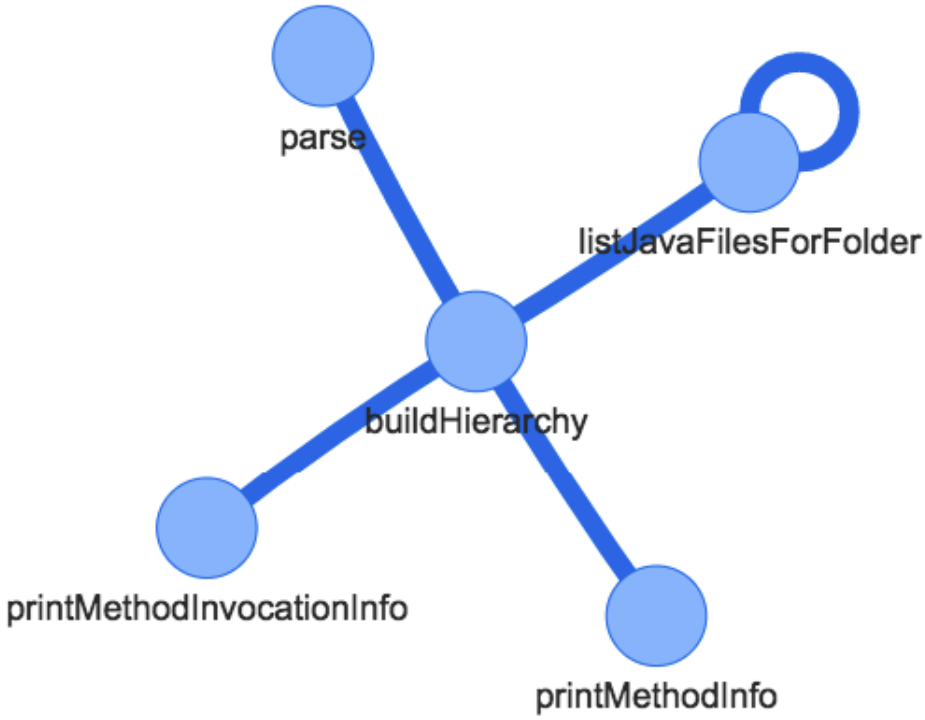
Ouvrir le client web dans un navigateur : http://localhost:2000

4 Résultat

4.1 Formulaire d'entrée



4.2 Graphe d'appel pour une classe



Note : Il est possible de déplacer les noeuds dans l'espace à l'aide du drag n drop

4.3 Graphe du projet